K  6

```
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
```

```
                                                  K  3
**FILE**ID**PCL11R

PPPPPPPP      CCCCCCCC  LL              11          11      RRRRRRRR
PPPPPPPP      CCCCCCCC  LL              11          11      RRRRRRRR
PP      PP  CC          LL            1111        1111      RR      RR
PP      PP  CC          LL            1111        1111      RR      RR
PP      PP  CC          LL              11          11      RR      RR
PP      PP  CC          LL              11          11      RR      RR
PPPPPPPP    CC          LL              11          11      RRRRRRRR
PPPPPPPP    CC          LL              11          11      RRRRRRRR
PP          CC          LL              11          11      RR  RR
PP          CC          LL              11          11      RR  RR
PP          CC          LL              11          11      RR    RR        ....
PP          CC          LL              11          11      RR    RR        ....
PP            CCCCCCCC  LLLLLLLLLL    111111      111111    RR      RR      ....
PP            CCCCCCCC  LLLLLLLLLL    111111      111111    RR      RR      ....


LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
C
C Version:       'V04-000'
C
C*************************************************************************
C*                                                                       *
C*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
C*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.               *
C*  ALL RIGHTS RESERVED.                                                 *
C*                                                                       *
C*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
C*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
C*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
C*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
C*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
C*  TRANSFERRED.                                                         *
C*                                                                       *
C*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
C*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
C*  CORPORATION.                                                         *
C*                                                                       *
C*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
C*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.              *
C*                                                                       *
C*                                                                       *
C*************************************************************************
C

c       Author  Brian Porter              Creation Date   31-MAY-1981


        Subroutine PCL11R (lun)

c++
c       Functional description:
c
c       This routine displays error log entries made by the
c       PCL11-B receiver driver.
c
c       Modified by:
c
C       V03-003 SAR0228         Sharon A. Reynolds,     28-Mar-1984
C               Changed the call to UCB$L_OWNUIC to ORB$L_OWNER.
C
C       V03-002 SAR0089         Sharon A. Reynolds,     20-Jun-1983
C               Changed the carriage control in the 'format' statements
C               for use with ERF.
C
C       V03-001 SAR0040         Sharon A. Reynolds,             8-Jun-1983
C               Removed brief/cryptic support.
C
c       v02-002 BP0002          Brian Porter,                   23-NOV-1981
c               Minor edit.
c
c       v02-001 BP0001          Brian Porter,                   30-SEP-1981
c               Corrected call to uba_mapping.
c               Added 'device attention' support.
c**
```

```
c--

        include 'src$:msghdr.for /nolist'
        include 'src$:deverr.for /nolist'


        byte            lun

        integer*4       rcr
        integer*4       rsr
        integer*4       rddb
        integer*4       rdbc
        integer*4       rdba
        integer*4       rdcrc
        integer*4       ucb$l_lr_dpn
        integer*4       ucb$l_lr_dpr
        integer*4       ucb$l_lr_fmpr
        integer*4       ucb$l_lr_pmpr
        integer*4       ucb$l_trigblk
        integer*4       ucb$l_devdepend

        equivalence     (emb$l_dv_regsav(0),rcr)
        equivalence     (emb$l_dv_regsav(1),rsr)
        equivalence     (emb$l_dv_regsav(2),rddb)
        equivalence     (emb$l_dv_regsav(3),rdbc)
        equivalence     (emb$l_dv_regsav(4),rdba)
        equivalence     (emb$l_dv_regsav(5),rdcrc)
        equivalence     (emb$l_dv_regsav(6),ucb$l_lr_dpn)
        equivalence     (emb$l_dv_regsav(7),ucb$l_lr_dpr)
        equivalence     (emb$l_dv_regsav(8),ucb$l_lr_fmpr)
        equivalence     (emb$l_dv_regsav(8),ucb$l_lr_pmpr)
        equivalence     (emb$l_dv_regsav(10),ucb$l_trigblk)
        equivalence     (emb$l_dv_regsav(11),ucb$l_devdepend)

        integer*4       compressc
        integer*4       compress4
        integer*4       response_bits
        integer*4       current_or_last_transmitter
        integer*4       selected_map_register

        character*26    v1rcr(0:7)
        data            v1rcr(0)        /'RECEIVE DATA*'/
        data            v1rcr(1)        /'RECEIVER INITIALIZE*'/
        data            v1rcr(2)        /'INHIBIT ADDRESS INCREMENT*'/
        data            v1rcr(3)        /'DATA INPUT READY*'/
        data            v1rcr(4)        /'EXTENDED BUS ADDRESS 16*'/
        data            v1rcr(5)        /'EXTENDED BUS ADDRESS 17*'/
        data            v1rcr(6)        /'INTERRUPT ENABLE*'/
        data            v1rcr(7)        /'LOAD SILO*'/

        character*13    v2rcr(13:15)
        data            v2rcr(13)       /'RECEIVE WORD*'/
        data            v2rcr(14)       /'RECEIVER NPR*'/
        data            v2rcr(15)       /'REJECT*'/

        character*22    v1rsr(4:15)
        data            v1rsr(4)        /'CHANNEL OPEN*'/
        data            v1rsr(5)        /'REJECT COMPLETE*'/
        data            v1rsr(6)        /'BUSY*'/
        data            v1rsr(7)        /'SUC TXF*'/
        data            v1rsr(8)        /'DATA OUTPUT READY*'/
        data            v1rsr(9)        /'BYTE COUNT OVERFLOW*'/
        data            v1rsr(10)       /'TIMEOUT*'/
        data            v1rsr(11)       /'PARITY*'/
        data            v1rsr(12)       /'TRANSMISSION ERROR*'/
        data            v1rsr(13)       /'MEMORY OVERFLOW*'/
        data            v1rsr(14)       /'NON-EXISTENT LOCATION*'/
        data            v1rsr(15)       /'ERROR*'/

        character*29    response_a(0:3)
        data            response_a(0)   /'TRANSMITTER ERROR/OFF-LINE*'/
        data            response_a(1)   /'NULL CYCLE*'/
        data            response_a(2)   /'VALID WORD/CRC ON DATA LINES*'/
        data            response_a(3)   /'LAST CRC ON DATA LINES*'/

        character*28    response_b(0:3)
        data            response_b(0)   /'RECEIVER ERROR/OFF-LINE*'/
        data            response_b(1)   /'NULL CYCLE*'/
        data            response_b(2)   /'CHECK FAILURE PREVIOUS DATA*'/
        data            response_b(3)   /'ACKNOWLEDGE CRC OR DATA*'/


c++
c       The format of the receiver driver buffer is as follows.
c
c       +-------------------------------------+
c       |                   rcr               |
c       +-------------------------------------+
c       |                   rsr               |
c       +-------------------------------------+
c       |              ucb$w_rddb             |
c       +-------------------------------------+
c       |                   rdbc              |
c       +-------------------------------------+
c       |                   rdba              |
c       +-------------------------------------+
c       |                   rdcrc             |
c       +-------------------------------------+
c       |              ucb$w_lr_dpn           |
c       +-------------------------------------+
c       |              ucb$l_lr_dpr           |
c       +-------------------------------------+
c       |              ucb$l_lr_fmpr          |
c       +-------------------------------------+
c       |              ucb$l_lr_pmpr          |
c       +-------------------------------------+
c       |              ucb$l_trigblk          |
c       +-------------------------------------+
c       |              ucb$l_devdepend        |
c       +-------------------------------------+
c       |              ucb$w_sts              |
c       +-------------------------------------+
c--

```

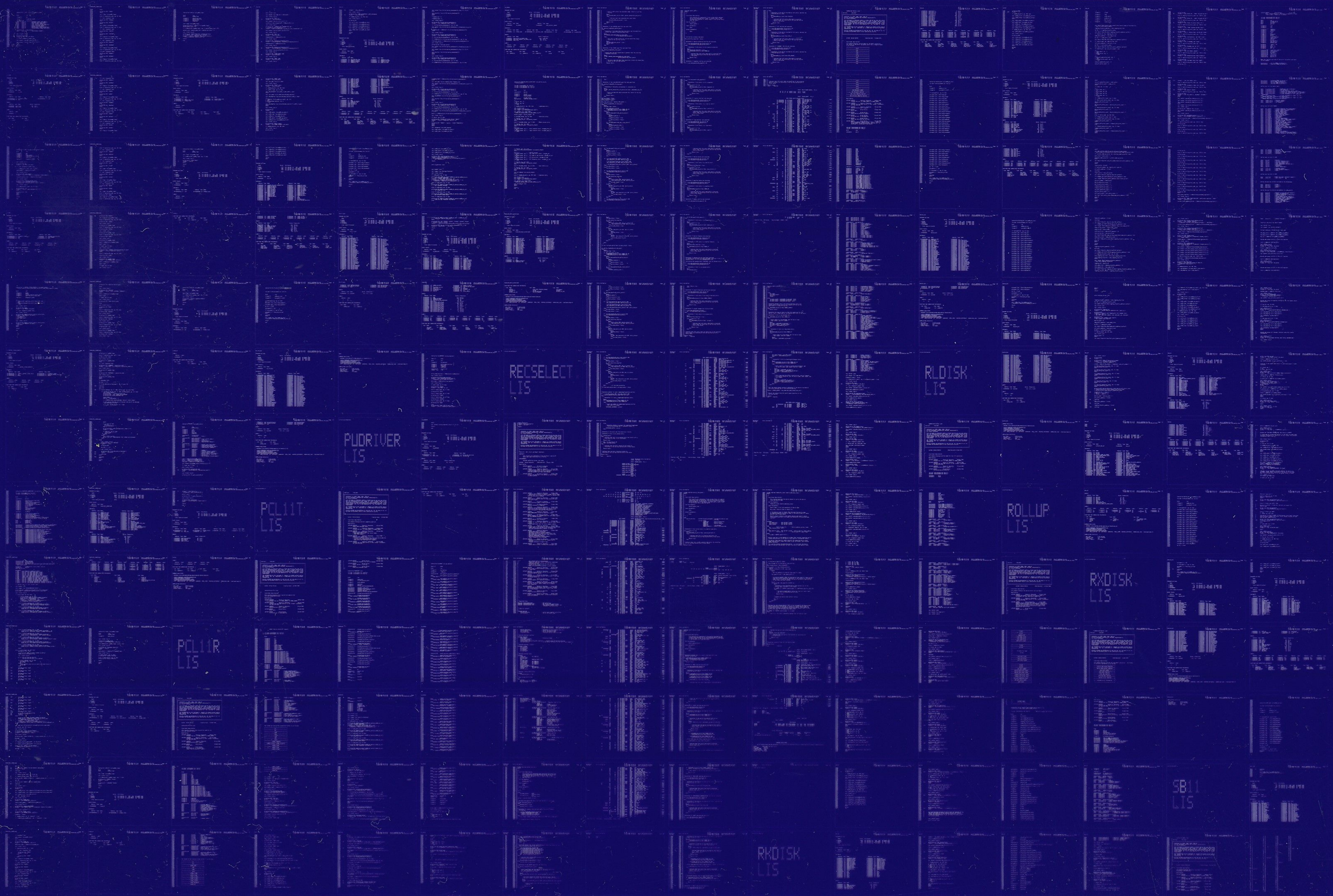
```



        call frctof (lun)

        call dhead1 (lun,'UBA PCL11 (RECEIVER)')

        call linchk (lun,2)

        write(lun,10) 'RCR',rcr
10      format(/' ',t8,a,t24,z8.4)

        call output (lun,rcr,v1rcr,0,0,7,'0')

        current_or_last_transmitter = lib$extzv(8,5,rcr)

        call linchk (lun,1)

        write(lun,15) 'CURRENT/LAST TRANSMITTER ID# ',
        1 current_or_last_transmitter,'.'
15      format(' ',t40,a,i<compress4 (current_or_last_transmitter)>,a)

        call output (lun,rcr,v2rcr,13,13,15,'0')

        call linchk (lun,1)

        write(lun,20) 'RSR',rsr
20      format(' ',t8,a,t24,z8.4)

        response_bits = lib$extzv(0,2,rsr)

        call linchk (lun,1)

        write(lun,25) response_a(response_bits)
25      format(' ',t40,a<compressc (response_a(response_bits))>)

        response_bits = lib$extzv(2,2,rsr)

        call linchk (lun,1)

        write(lun,30) response_b(response_bits)
30      format(' ',t40,a<compressc (response_b(response_bits))>)

        call output (lun,rsr,v1rsr,4,4,15,'0')

        call linchk (lun,1)

        write(lun,35) 'UCB$W_RDDB',rddb
35      format(' ',t8,a,t24,z8.4)

        call linchk (lun,1)

        write(lun,40) 'RDBC',rdbc
40      format(' ',t8,a,t24,z8.4)

        call linchk (lun,1)

        write(lun,45) 'RDBA',rdba
45      format(' ',t8,a,t24,z8.4)

        call calc_map (lun,4,rcr,rdba)

        call linchk (lun,1)

        write(lun,50) 'RDCRC',rdcrc
50      format(' ',t8,a,t24,z8.4)

        if (emb$w_hd_entry .ne. 98) then

        if (ucb$l_lr_dpn .ne. 0) then

        call uba_datapath (lun,ucb$l_lr_dpn,ucb$l_lr_dpr)
        endif

        call calc_map2 (4,rcr,rdba,selected_map_register)

        call uba_mapping (lun,selected_map_register,ucb$l_lr_fmpr)

        if (
        1 lib$extzv (16,16,emb$l_dv_iosb1) .gt. 512
        1 .and.
        1 selected_map_register .ne. 0
        1 ) then

        call uba_mapping (lun,(selected_map_register-1),ucb$l_lr_pmpr)
        endif
        endif

        call linchk (lun,1)

        write(lun,55)
55      format(' ',:)

        call orb$l_owner (lun,emb$l_dv_ownuic)

        call ucb$l_char (lun,emb$l_dv_char)

        call ucb$w_sts (lun,emb$w_dv_sts)

        call ucb$l_opcnt (lun,emb$l_dv_opcnt)

        call ucb$w_errcnt (lun,emb$w_dv_errcnt)

        call linchk (lun,1)

        write(lun,65) ucb$l_trigblk
65      format(' ',t8,'UCB$L_TRIGBLK',t24,z8.8)

        if (emb$w_hd_entry .ne. 98) then

        call linchk (lun,1)

        write(lun,55)

        call pcl11r_qio (lun,emb$w_dv_func)

        call irp$w_bcnt (lun,emb$w_dv_bcnt)

        call irp$w_boff (lun,emb$w_dv_boff)

        call irp$l_pid (lun,emb$l_dv_rqpid)

        call irp$q_iosb (lun,emb$l_dv_iosb1)
        endif

        return
        end


PROGRAM SECTIONS

    Name                                        Bytes   Attributes

  0 $CODE                                        1175   PIC CON REL LCL   SHR   EXE   RD NOWRT LONG
  1 $PDATA                                        264   PIC CON REL LCL   SHR NOEXE   RD NOWRT LONG
  2 $LOCAL                                       1388   PIC CON REL LCL NOSHR NOEXE   RD   WRT LONG
  3 EMB                                           512   PIC OVR REL GBL   SHR NOEXE   RD   WRT LONG

    Total Space Allocated                        3339

ENTRY POINTS

    Address  Type  Name

  0-00000000       PCL11R

VARIABLES

    Address  Type  Name                                  Address  Type  Name

  2-000002E8  I*4  CURRENT_OR_LAST_TRANSMITTER        3-0000001C  L*1  EMB$B_DV_CLASS
  3-00000010  L*1  EMB$B_DV_ERTCNT                    3-00000011  L*1  EMB$B_DV_ERTMAX
  3-0000003E  L*1  EMB$B_DV_NAMLNG                    3-0000003A  L*1  EMB$B_DV_SLAVE
  3-0000001D  L*1  EMB$B_DV_TYPE                      3-00000036  I*4  EMB$L_DV_CHAR
  3-00000012  I*4  EMB$L_DV_IOSB1                     3-00000016  I*4  EMB$L_DV_IOSB2
  3-00000026  I*4  EMB$L_DV_MEDIA                     3-0000004E  I*4  EMB$L_DV_NUMREG
  3-0000002E  I*4  EMB$L_DV_OPCNT                     3-00000032  I*4  EMB$L_DV_OWNUIC
  3-0000001E  I*4  EMB$L_DV_RQPID                     3-00000000  I*4  EMB$L_HD_SID
  3-0000003F  CHAR EMB$T_DV_NAME                      3-00000024  I*2  EMB$W_DV_BCNT
  3-00000022  I*2  EMB$W_DV_BOFF                      3-0000002C  I*2  EMB$W_DV_ERRCNT
  3-0000003C  I*2  EMB$W_DV_FUNC                      3-0000001A  I*2  EMB$W_DV_STS
  3-0000002A  I*2  EMB$W_DV_UNIT                      3-00000004  I*2  EMB$W_HD_ENTRY
  3-0000000E  I*2  EMB$W_HD_ERRSEQ                   AP-00000004@ L*1  LUN
  3-00000052  I*4  RCR                                3-00000062  I*4  RDBA
  3-0000005E  I*4  RDBC                               3-00000066  I*4  RDCRC
  3-0000005A  I*4  RDDB                               2-000002E4  I*4  RESPONSE_BITS
  3-00000056  I*4  RSR                                2-000002EC  I*4  SELECTED_MAP_REGISTER
  3-0000007E  I*4  UCB$L_DEVDEPEND                              3-0000006A  I*4  UCB$L_LR_DPN
  3-0000006E  I*4  UCB$L_LR_DPR                                 3-00000072  I*4  UCB$L_LR_FMPR
  3-00000072  I*4  UCB$L_LR_PMPR                                3-0000007A  I*4  UCB$L_TRIGBLK

ARRAYS

    Address  Type  Name                                   Bytes  Dimensions

  3-00000000  L*1  EMB                                      512  (0:511)
  3-00000052  I*4  EMB$L_DV_REGSAV                          420  (0:104)
  3-00000006  I*4  EMB$Q_HD_TIME                              8  (2)
  2-000001FF  CHAR RESPONSE_A                               116  (0:3)
  2-00000273  CHAR RESPONSE_B                               112  (0:3)
  2-00000000  CHAR V1RCR                                    208  (0:7)
  2-000000F7  CHAR V1RSR                                    264  (4:15)
  2-000000D0  CHAR V2RCR                                     39  (13:15)

LABELS

    Address   Label       Address   Label       Address   Label       Address   Label       Address   Label       Address   Label

  1-0000007A  10'     1-00000087  15'     1-00000095  20'     1-000000A1  25'     1-000000AD  30'     1-000000B9  35'
  1-000000C5  40'     1-000000D1  45'     1-000000DD  50'     1-000000E9  55'     1-000000EE  65'

FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name            Type  Name            Type  Name            Type  Name            Type  Name            Type  Name

        CALC_MAP              CALC_MAP2       I*4   COMPRESS4       I*4   COMPRESSC             DHEAD1                FRCTOF
        IRP$L_PID             IRP$Q_IOSB            IRP$W_BCNT            IRP$W_BOFF      I*4   LIB$EXTZV             LINCHK
        ORB$L_OWNER           OUTPUT                PCL11R_QIO            UBA_DATAPATH          UBA_MAPPING           UCB$L_CHAR
        UCB$L_OPCNT           UCB$W_ERRCNT          UCB$W_STS
```

```
      Subroutine PCL11R_QIO (lun,emb$w_dv_func)


      include 'src$:qiocommon.for /nolist'

      byte              lun

      integer*2         emb$w_dv_func

      integer*4         qiocode(0:1,0:63)

      if (qiocode(0,0) .eq. 0) then

      qiocode(1,12) = %loc(io$_readpblk)

      qiocode(1,33) = %loc(io$_readlblk)

      do 10,i = 0,63

      qiocode(0,i) = 33

      if (qiocode(1,i) .eq. 0) then

      qiocode(1,i) = %loc(qio_string)
      endif

10    continue
      endif

      call irp$w_func (lun,emb$w_dv_func,
     1 qiocode(0,lib$extzv(0,6,emb$w_dv_func)))

      return

      end
```

```
PROGRAM SECTIONS

    Name                                  Bytes   Attributes

  0 $CODE                                   111   PIC CON REL LCL   SHR   EXE   RD NOWRT LONG
  1 $PDATA                                    8   PIC CON REL LCL   SHR NOEXE   RD NOWRT LONG
  2 $LOCAL                                  548   PIC CON REL LCL NOSHR NOEXE   RD   WRT LONG
  3 QIOCOMMON                              1247   PIC OVR REL GBL   SHR NOEXE   RD   WRT LONG

    Total Space Allocated                  1914

ENTRY POINTS

    Address   Type  Name

  0-00000000        PCL11R_QIO

VARIABLES

    Address   Type  Name                                   Address   Type  Name

 AP-00000008@ I*2   EMB$W_DV_FUNC                        2-00000200   I*4  I
  3-00000442  CHAR  IO$_ABORT                            3-0000034D   CHAR IO$_ACCESS
  3-000003C2  CHAR  IO$_ACPCONTROL                       3-000004B3   CHAR IO$_AVAILABLE
  3-00000297  CHAR  IO$_CLEAN                            3-00000369   CHAR IO$_CREATE
  3-00000385  CHAR  IO$_DEACCESS                         3-00000393   CHAR IO$_DELETE
  3-0000026D  CHAR  IO$_DIAGNOSE                         3-00000065   CHAR IO$_DRVCLR
  3-000004CB  CHAR  IO$_DSE                              3-000000A9   CHAR IO$_ERASETAPE
  3-00000276  CHAR  IO$_FORMAT                           3-00000071   CHAR IO$_INITIALIZE
  3-00000014  CHAR  IO$_LOADMCODE                        3-000003A1   CHAR IO$_MODIFY
  3-000003E2  CHAR  IO$_MOUNT                            3-00000000   CHAR IO$_NOP
  3-0000009D  CHAR  IO$_OFFSET                           3-000000EB   CHAR IO$_PACKACK
  3-000000E0  CHAR  IO$_QSTOP                            3-000003EF   CHAR IO$_RDSTATS
  3-00000421  CHAR  IO$_READCSR                          3-00000169   CHAR IO$_READHEAD
  3-000002B6  CHAR  IO$_READLBLK                         3-0000013F   CHAR IO$_READPBLK
  3-00000200  CHAR  IO$_READPRESET                       3-00000195   CHAR IO$_READTRACKD
  3-0000033A  CHAR  IO$_READVBLK                         3-0000045A   CHAR IO$_READWTHBUF
  3-00000484  CHAR  IO$_READWTHXBUF                      3-0000004D   CHAR IO$_RECAL
  3-0000007C  CHAR  IO$_RELEASE                          3-000001AB   CHAR IO$_REREADN
  3-000001B8  CHAR  IO$_REREADP                          3-000000CA   CHAR IO$_RETCENTER
  3-000002E6  CHAR  IO$_REWIND                           3-000002C9   CHAR IO$_REWINDOFF
  3-000000FC  CHAR  IO$_SEARCH                           3-00000024   CHAR IO$_SEEK
  3-00000231  CHAR  IO$_SENSECHAR                        3-00000309   CHAR IO$_SENSEMODE
  3-0000021D  CHAR  IO$_SETCHAR                          3-000003B8   CHAR IO$_SETCLOCK
  3-00000088  CHAR  IO$_SETCLOCKP                        3-000002DD   CHAR IO$_SETMODE
  3-000002ED  CHAR  IO$_SKIPFILE                         3-000002FA   CHAR IO$_SKIPRECORD
  3-00000029  CHAR  IO$_SPACEFILE                        3-0000010E   CHAR IO$_SPACERECORD
  3-000003D7  CHAR  IO$_STARTDATA                        3-000000B4   CHAR IO$_STARTDATAP
  3-00000037  CHAR  IO$_STARTMPROC                       3-0000020F   CHAR IO$_STARTSPNDL
  3-00000059  CHAR  IO$_STOP                             3-0000000D   CHAR IO$_UNLOAD
  3-0000046B  CHAR  IO$_WRITEBUFNCRC                     3-0000011E   CHAR IO$_WRITECHECK
  3-000001E4  CHAR  IO$_WRITECHECKH                      3-000003FF   CHAR IO$_WRITECSR
  3-00000153  CHAR  IO$_WRITEHEAD                        3-000002A2   CHAR IO$_WRITELBLK
  3-00000247  CHAR  IO$_WRITEMARK                        3-00000314   CHAR IO$_WRITEOF
  3-0000012A  CHAR  IO$_WRITEPBLK                        3-000001C9   CHAR IO$_WRITERET
  3-0000017E  CHAR IO$_WRITETRACKD                          3-00000326  CHAR IO$_WRITEVBLK
  3-00000448  CHAR IO$_WRITEWTHBUF                          3-00000257  CHAR IO$_WRTTMKR
 AP-00000004@ L*1  LUN                                      3-000004A1  CHAR QIO_STRING

ARRAYS

    Address  Type  Name                    Bytes  Dimensions

  2-00000000  I*4  QIOCODE                   512  (0:1, 0:63)

LABELS

    Address   Label

       **     10

FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name              Type  Name

        IRP$W_FUNC         I*4  LIB$EXTZV

COMMAND QUALIFIERS

  FORTRAN /LIS=LIS$:PCL11R/OBJ=OBJ$:PCL11R MSRC$:PCL11R

  /CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /STANDARD=(NOSYNTAX,NOSOURCE_FORM)
  /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)
  /F77  /NOG_FLOATING  /I4  /OPTIMIZE  /WARNINGS  /NOD_LINES  /NOCROSS_REFERENCE  /NOMACHINE_CODE  /CONTINUATIONS=19

COMPILATION STATISTICS

  Run Time:             5.69 seconds
  Elapsed Time:         17.88 seconds
  Page Faults:          187
  Dynamic Memory:       193 pages
```

0153 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



RECSELECT
LIS

RLDISK
LIS

PUDRIVER
LIS

PCL11T
LIS

ROLLUP
LIS

RXDISK
LIS

PCL11R
LIS

SB11
LIS

RKDISK
LIS